REMARKS

This paper is responsive to the Office action mailed March 6, 2003. Claims 9-36 were examined. Claims 28, 31 and 33 were objected to based on minor informalities that have been addressed by amendment. Claims 9-36 were all rejected under 35 U.S.C. § 102(e) over U.S. Patent 5,488,729 to Vegesna et al (*Vegesna*). The § 102 rejections are traversed.

Applicant appreciates the Examiner's thorough review of specification, claims and drawings. Appropriate corrections have been made as requested by the Examiner.

Art Rejections – 35 U.S.C. § 102(e)

Applicant respectfully suggests that the Office has misinterpreted *Vegesna* in at least two important fundamental ways. First, *Vegesna* discloses a simple, single-pipeline-stage technique in which instruction decode and scheduling, including *interpacket* and *intrapacket* scheduling are performed in a single pipeline stage. *See* discussion throughout *Vegesna* of pipeline stage D. "CSCHED 2 performs these tasks [(decode and scheduling)] in parallel so that they will be completed ***within one CPU clock cycle***." See *Vegesna*, col. 26, lines 7-14.

While this approach may be practical or acceptable in the 2-wide packets and minimally pipelined architecture of *Vegesna*, scaling to architectures that provide wider dispatch width is quite poor. As pointed out in Applicant's disclosure, as a general matter, as the number of functional units that can be simultaneously executed increases, time required for scheduling (e.g., resource checking and data dependency checking) scales more dramatically than linearly. *See* Application, Background and Summary p. 2, line 36 through p. 3, line 33. Therefore, using a single-stage, single-cycle decode/scheduling pipeline such as disclosed in *Vegesna*, complexity of the decode/scheduling logic scales non-linearly as packet width increases and that complexity becomes a limiting factor on processor cycle time.

Accordingly, Applicant has developed techniques whereby instruction grouping for dispatch (including intra-group and inter-group dependency checking) can be performed over multiple pipeline stages, i.e., spanning multiple processor cycles. In this way, at least some of the limiting effects of increased grouping (or scheduling) complexity on processor cycle time for wide-dispatch width processors can be avoided.

Turning to the claim language, claim 1 recites a processor that:

> performs, over plural execution cycles ..., instruction grouping for dispatch, including both intra-group and inter-group dependency checking.

As described above, *Vegesna* simply does not perform its grouping and dependency checks over plural cycles. Indeed, quite to the contrary, *Vegesna* performs its analogous scheduling operations in a single pipeline stage (stage D) so that they will be completed ***within one CPU clock cycle***. For at least this reason, claim 9 and those dependent therefrom (claims 10-16) are allowable over *Vegesna* and a notice to that effect is respectfully requested.

Though of substantially differing scope, independent claims 17, 26, 31 and 35 each recite limitations related to performance of analogous functions over plural cycles and/or pipeline stages. Accordingly, each together with claims dependent therefrom (i.e., claims 17-36) is allowable over *Vegesna* for at least the reasons given above. A notice to that effect is respectfully requested.

A second fundamental misinterpretation of *Vegesna* presents itself most clearly in the form of the Office's rejection of claims 12, 26 and 32. In particular, the Office incorrectly states (quoting Applicant's claim language) that Vegesna "dispatches all instructions from a particular one of the successive groups before dispatching any instructions from a subsequent one of the successive groups." *See* Office action, ¶ 14 (citing *Vegesna*, col. 22, lines 40-49). In point of fact, *Vegesna*, particularly col. 22, lines 40-49 thereof is quite to the contrary. Indeed, that section of *Vegesna* describes that an unissued instruction in "SLOTB" of a given *packet* (DBUF) is shifted into the "SLOTA" position thereof and another valid instruction (i.e., a SLOTA instruction from the *subsequent packet* in FBUF) fills the SLOTB position of DBUF. In short, *Vegesna* merges an instruction from a subsequent packet into a preceding packet. SLOTA and SLOTB instructions can then be dispatched together. As a result, if *packets* are equivalent to *groups* (as apparently assumed by the Office), *Vegesna* does not:

> .. dispatch[] all instructions from a particular one of the successive groups before dispatching any instructions from a subsequent one of the successive groups.

As recited in claim 12. Though of substantially differing scope, claims 26 and 32 each recite limitations related to non-merger of instruction groups, and are each allowable for the same or similar reasons.

In summary, claims 9-36 are pending. All claims are believed to be allowable over the art of record, and a Notice of Allowance to that effect is respectfully solicited. Nonetheless, if any issues remain that could be more efficiently handled by telephone, the Examiner is requested to call the undersigned at the number listed below.

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that, on the date shown below, this correspondence is being

☐ deposited with the US Postal Service with sufficient postage as first class mail in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

☒ facsimile transmitted to the US Patent and Trademark Office.

8/6/03
Date

EXPRESS MAIL LABEL: _____

Respectfully submitted,

David W. O'Brien, Reg. No. 40,107
Attorney for Applicant(s)
(512) 347-9030
(512) 347-9031 (fax)